

## EXHIBIT E

### Refresh Menu Option

```
Private Sub mnuDSMenuOptions_Click(Index As Integer)
    Dim SQL As String
    Dim Msg As String
    Dim i As Long
    Dim RefreshForm As Boolean
    Dim CurrentID As Long

    Select Case tvwData.SelectedItem.Text
        Case "Datasources"
            Select Case Index
                Case 0 'New
                    SetMessage "Loading datasource properties..."
                    LockWindow True
                    frmDatasource.DatasourceID = -1
                    frmDatasource.LoadForm
                    SetMessage gData.DatabasePath
                    frmDatasource.Show vbModal
                    Me.Refresh
                    If Not frmDatasource.Cancelled Then
                        tvwData_NodeClick tvwData.SelectedItem
                        RefreshDatabaseConnections
                    End If
                    Unload frmDatasource
                Case 2 'Delete
                    Msg = "Are you sure you want to delete the selected datasource(s)?"
                    If MsgBox(Msg, vbYesNo + vbQuestion, "Delete Datasource(s)") = vbYes Then
                        RefreshForm = False
                        For i = 1 To lstData.ListItems.Count
                            If lstData.ListItems(i).Selected = True Then
                                SQL = "Delete From FW_DATASOURCE Where DATASOURCE_ID=" & lstData.ListItems(i).Tag
                                If Not RefreshForm Then
                                    RefreshForm = DLookup("DATA_ENTRY", "FW_DATASOURCE", "DATASOURCE_ID=" &
                                        lstData.ListItems(i).Tag)
                                End If
                                Call ExecuteSQL(SQL)
                            End If
                        Next i
                        If RefreshForm Then
                            LoadForm
                        Else
                            tvwData_NodeClick tvwData.SelectedItem
                        End If
                    End If
                Case 4 'Properties
                    SetMessage "Loading datasource properties..."
                    LockWindow True
                    CurrentID = lstData.SelectedItem.Tag
                    frmDatasource.DatasourceID = CurrentID
                    frmDatasource.LoadForm
                    SetMessage gData.DatabasePath
                    frmDatasource.Show vbModal
                    Me.Refresh
                    If Not frmDatasource.Cancelled Then
                        tvwData_NodeClick tvwData.SelectedItem
                        If Not lstData.SelectedItem Is Nothing Then
                            lstData.SelectedItem.Selected = False
                        End If
                        Set lstData.SelectedItem = lstData.FindItem(CurrentID, lstData.Tag)
                        RefreshDatabaseConnections
                    End If
                    Unload frmDatasource
                Case 6 'Refresh
                    If gData.ValidDataEntryConnection = False Then
                        NoConnectionMessage
                        Exit Sub
                    End If
                    Me.Enabled = False
                    RefreshDatasource Me, lstData.SelectedItem.Tag
                    Me.Enabled = True
                    LoadForm
            End Select
        End Select
    End Sub
```

## EXHIBIT E

### RefreshDatasource

```
Public Sub RefreshDatasource(f As Form, DatasourceID As Long)
    Dim db As JRO.JetEngine
    Dim dDef As ADOX.Catalog
    Dim tDef As ADOX.Table
    Dim DualTable As String
    Dim AuditTable As String
    Dim rs As ADODB.Recordset
    Dim rsTables As ADODB.Recordset
    Dim SQL As String
    Dim EntityID As Long
    Dim EntityColumnID As Long
    Dim i As String
    Dim cPerm As CPermission
    Dim TableViewPrefix As String
    Dim con As ADODB.Connection
    Dim InitialLoad As Boolean
    Dim DisplayOnViewMenu As Boolean

    If gData.ValidDataEntryConnection = False Then
        NoConnectionMessage
        Exit Sub
    End If
    TableViewPrefix = DLookup("DATASOURCE_ATTRIBUTE_VALUE", "DATASOURCE_ATTRIBUTE_V",
"DATASOURCE_ATTRIBUTE='INITIAL_CATALOG' AND DATASOURCE_ID=" & DatasourceID) & ""
    f.SetMessage "Loading database tables..."
    LockWindow True
    DisplayOnViewMenu = False
    InitialLoad = IsNull(DLookup("ENTITY_ID", "FW_ENTITY"))
    frmReferenceTables.InitialLoad = InitialLoad
    frmReferenceTables.DatasourceID = DatasourceID
    f.SetMessage gData.DatabasePath
    frmReferenceTables.Show vbModal
    If frmReferenceTables.Cancelled Then
        Unload frmReferenceTables
        Exit Sub
    Else
        If InitialLoad Then
            DisplayOnViewMenu = frmReferenceTables.DisplayOnViewMenu
        End If
        Unload frmReferenceTables
    End If

    f.SetMessage "Refreshing datasource..."
    f.Refresh
    DoEvents

    LockWindow True

    Set dDef = New ADOX.Catalog
    Set dDef.ActiveConnection = GetDatasourceConnection(DatasourceID)
    If dDef.ActiveConnection Is Nothing Then
        Set dDef = Nothing
        LockWindow False
        f.SetMessage gData.DatabasePath
        Exit Sub
    End If

    VerifySequenceTable dDef, DatasourceID, TableViewPrefix

    DualTable = DLookup("DUAL_TABLE", "FW_DATASOURCE", "DATASOURCE_ID=" & DatasourceID)
    VerifyDualTable dDef, DualTable, DatasourceID
    InsertDummyRecord DatasourceID, DualTable

    AuditTable = DLookup("AUDIT_TABLE", "FW_DATASOURCE", "DATASOURCE_ID=" & DatasourceID)
    VerifyAuditTable dDef, AuditTable, DatasourceID, TableViewPrefix

    Set con = dDef.ActiveConnection
    Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
        Case "ACCESS.OLEDB"
            Set rsTables = con.OpenSchema(adSchemaTables, Array(Empty, Empty, Empty, Empty))
        Case "ORACLE.OLEDB"
            Set rsTables = con.OpenSchema(adSchemaTables, Array(Empty, TableViewPrefix, Empty, Empty))
        Case "SQLSERVER.OLEDB"
            Set rsTables = con.OpenSchema(adSchemaTables, Array(TableViewPrefix, Empty, Empty, Empty))
    End Select

    Do While Not rsTables.EOF
```

## EXHIBIT E

```
If (rsTables("TABLE_TYPE").Value = "TABLE") And (rsTables("TABLE_NAME").Value <> "SEQUENCE") _
And (rsTables("TABLE_NAME").Value <> DualTable) And (rsTables("TABLE_NAME").Value <> AuditTable) Then
Set tDef = dDef.Tables(rsTables("TABLE_NAME").Value)
SQL = "Select * From FW_ENTITY Where TABLE_NAME='" & tDef.Name & "'"
Set rs = GetEditRS(SQL)
If rs.BOF And rs.EOF Then
    rs.AddNew
    rs("TABLE_NAME").Value = tDef.Name
    rs("DATASOURCE_ID").Value = DatasourceID
    rs("ENTITY_CAPTION").Value = StrConv(Replace(tDef.Name, "_", " ", 1, -1, vbTextCompare), vbProperCase)
    rs("UNIQUE_FIELD_NAME").Value = GetPrimaryKey(tDef)
    rs("SHOW_UNIQUE_FIELD").Value = False
    rs("RECORD_SOURCE").Value = GetRecordsource(dDef, tDef.Name)
    rs("VISIBLE").Value = True
    rs("REFERENCE").Value = IsReferenceTables(tDef.Name)
    If rs("REFERENCE").Value Then
        rs("DISPLAY_ON_MENU").Value = False
        rs("ICON").Value = Null
    Else
        rs("DISPLAY_ON_MENU").Value = DisplayOnViewMenu
        If DisplayOnViewMenu Then
            rs("ICON").Value = GetINISetting(App.Path & "\SystemConsole.ini", "System Settings", "Default Entity
Icon", "ERROR")
        Else
            rs("ICON").Value = Null
        End If
    End If
    rs("RELATIVE_ORDER").Value = Null
    rs.Update
    EntityID = rs("ENTITY_ID").Value
    Set cPerm = New CPermission
    cPerm.PrimaryKeyID = -1
    cPerm.CurrentID = EntityID
    cPerm.DatasourceID = DatasourceID
    If rs("REFERENCE").Value = True Then
        cPerm.MenuContext = "REFERENCE"
    Else
        cPerm.MenuContext = "ENTITY"
    End If
    cPerm.CreatePermissions
    Set cPerm = Nothing
Else
    If rs("REFERENCE").Value Then 'Currently reference table
        If Not IsReferenceTables(tDef.Name) Then
            rs("ICON").Value = GetINISetting(App.Path & "\SystemConsole.ini", "System Settings", "Default Entity
Icon", "ERROR")
            rs("DISPLAY_ON_MENU").Value = False
            rs("REFERENCE").Value = False
            rs.Update

            SQL = "Delete From FW_MENU_ITEM_PERMISSION Where ENTITY_GROUP_ID In (Select ENTITY_GROUP_ID "
            SQL = SQL & "From FW_ENTITY_GROUP Where ENTITY_ID=" & rs("ENTITY_ID").Value & ")"
            Call ExecutesSQL(SQL)

            SQL = "Delete From FW_COMMAND_OPTION_PERMISSION Where ENTITY_GROUP_ID In (Select ENTITY_GROUP_ID "
            SQL = SQL & "From FW_ENTITY_GROUP Where ENTITY_ID=" & rs("ENTITY_ID").Value & ")"
            Call ExecutesSQL(SQL)

            SQL = "Delete From FW_ENTITY_GROUP Where ENTITY_ID=" & rs("ENTITY_ID").Value
            Call ExecutesSQL(SQL)

            Set cPerm = New CPermission
            cPerm.PrimaryKeyID = -1
            cPerm.CurrentID = rs("ENTITY_ID").Value
            cPerm.DatasourceID = DatasourceID
            cPerm.MenuContext = "ENTITY"
            cPerm.CreatePermissions
            Set cPerm = Nothing

            ModifyRelationships rs("ENTITY_ID").Value, False
        End If
    Else 'Not currently a reference table.
        If IsReferenceTables(tDef.Name) Then
            rs("ICON").Value = Null
            rs("DISPLAY_ON_MENU").Value = False
            rs("RELATIVE_ORDER").Value = Null
            rs("REFERENCE").Value = True
            rs.Update
        End If
    End If
End If
```

## EXHIBIT E

```
SQL = "Delete From FW_MENU_ITEM_PERMISSION Where ENTITY_GROUP_ID In (Select ENTITY_GROUP_ID "
SQL = SQL & "From FW_ENTITY_GROUP Where ENTITY_ID=" & rs("ENTITY_ID").Value & ")"
Call ExecuteSQL(SQL)

SQL = "Delete From FW_COMMAND_OPTION_PERMISSION Where ENTITY_GROUP_ID In (Select ENTITY_GROUP_ID "
SQL = SQL & "From FW_ENTITY_GROUP Where ENTITY_ID=" & rs("ENTITY_ID").Value & ")"
Call ExecuteSQL(SQL)

SQL = "Delete From FW_ENTITY_GROUP Where ENTITY_ID=" & rs("ENTITY_ID").Value
Call ExecuteSQL(SQL)

Set cPerm = New CPermission
cPerm.PrimaryKeyID = -1
cPerm.CurrentID = rs("ENTITY_ID").Value
cPerm.DatasourceID = DatasourceID
cPerm.MenuContext = "REFERENCE"
cPerm.CreatePermissions
Set cPerm = Nothing

    ModifyRelationships rs("ENTITY_ID").Value, True
End If
End If
End If
rs.Close: Set rs = Nothing
End If
rsTables.MoveNext
Loop
rsTables.Close: Set rsTables = Nothing
OrderEntities
CreateRelationships
RefreshEntities

LoadSequences
RefreshContexts
Set dDef = Nothing
On Error Resume Next
Set db = New JRO.JetEngine
db.CompactDatabase GetConnectionString(), GetTempConnectionString()
Set db = Nothing
If Err.Number = 0 Then
    FileCopy gData.DatabasePath & "~", gData.DatabasePath
    Kill gData.DatabasePath & "~"
End If
Err.Clear
LockWindow False
f.SetMessage gData.DatabasePath

End Sub
```

## EXHIBIT E

### VerifySequenceTable

```
Private Sub VerifySequenceTable(dDef As ADOX.Catalog, DatasourceID As Long, TableViewPrefix As String)
    Dim tDef As ADOX.Table
    Dim SQL As String

    On Error Resume Next
    Set tDef = dDef.Tables("SEQUENCE")
    If Err.Number <> 0 Then
        Err.Clear
        Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
            Case "ACCESS.OLEDB"
                Set tDef = New ADOX.Table
                tDef.Name = "SEQUENCE"
                tDef.Columns.Append "TABLE_NAME", adWChar, 50
                tDef.Columns.Append "NEXT_VAL", adInteger
                dDef.Tables.Append tDef
            Case "SQLSERVER.OLEDB"
                Set tDef = New ADOX.Table
                tDef.Name = "SEQUENCE"
                tDef.Columns.Append "TABLE_NAME", adVarChar, 50
                tDef.Columns.Append "NEXT_VAL", adInteger
                dDef.Tables.Append tDef
            Case "ORACLE.OLEDB"
                SQL = "CREATE TABLE " & TableViewPrefix & ".SEQUENCE" & vbCrLf
                SQL = SQL & "(" & vbCrLf
                SQL = SQL & "TABLE_NAME VARCHAR2(50) NOT NULL," & vbCrLf
                SQL = SQL & "NEXT_VAL NUMBER(6,0) NOT NULL" & vbCrLf
                SQL = SQL & ")" & vbCrLf
                SQL = SQL & "TABLESPACE SYSTEM"
                Call ExecuteDataSourceSQL(DatasourceID, SQL)
        End Select
        Err.Clear
    End If
End Sub
```

**EXHIBIT E**  
**VerifyDualTable**

```
Private Sub VerifyDualTable(dDef As ADOX.Catalog, TableName As String, DatasourceID As Long)
    Dim tDef As ADOX.Table

    On Error Resume Next
    Set tDef = dDef.Tables(TableName)
    If Err.Number <> 0 Then
        Err.Clear
        Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
            Case "ACCESS.OLEDB"
                Set tDef = New ADOX.Table
                tDef.Name = TableName
                tDef.Columns.Append "DUMMY_FIELD", adWChar, 50
                dDef.Tables.Append tDef
            Case "SQLSERVER.OLEDB"
                Set tDef = New ADOX.Table
                tDef.Name = TableName
                tDef.Columns.Append "DUMMY_FIELD", adVarChar, 50
                dDef.Tables.Append tDef
            Case "ORACLE.OLEDB"
                'No action needed for Oracle
        End Select
        Err.Clear
    End If
End Sub
```

**EXHIBIT E**  
InsertDummyRecord

```
Private Sub InsertDummyRecord(DatasourceID As Long, TableName As String)
    Dim rs As ADODB.Recordset
    Dim com As ADODB.Command

    Set com = New ADODB.Command

    Set com.ActiveConnection = GetDataSourceConnection(DatasourceID)
    If com.ActiveConnection Is Nothing Then
        Set com = Nothing
        Exit Sub
    End If

    On Error Resume Next
    com.CommandType = adCmdText
    com.CommandText = "Select * From " & TableName
    com.CommandTimeout = 300

    Set rs = New ADODB.Recordset
    rs.CursorType = adOpenStatic
    rs.CursorLocation = adUseClient
    rs.LockType = adLockOptimistic
    rs.Open com
    If Err.Number = 0 Then
        If rs.BOF And rs.EOF Then
            rs.AddNew
            rs(0).Value = "Dummy"
            rs.Update
        End If
    End If
    Err.Clear
    Set com = Nothing
    rs.Close: Set rs = Nothing
End Sub
```

## EXHIBIT E

### VerifyAuditTable

```
Private Sub VerifyAuditTable(dDef As ADOX.Catalog, TableName As String, DatasourceID As Long, TableViewPrefix As String)
    Dim tDef As ADOX.Table
    Dim SQL As String

    On Error Resume Next
    Set tDef = dDef.Tables(TableName)
    If Err.Number <> 0 Then
        Err.Clear
        Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
            Case "ACCESS.OLEDB"
                Set tDef = New ADOX.Table
                tDef.Name = TableName
                tDef.Columns.Append "TABLE_NAME", adWChar, 50
                tDef.Columns.Append "USER_NAME", adWChar, 20
                tDef.Columns.Append "ACTION_TYPE", adWChar, 1
                tDef.Columns.Append "AUDIT_TIMESTAMP", adDate
                tDef.Columns.Append "RECORD_ID", adInteger
                dDef.Tables.Append tDef
            Case "ORACLE.OLEDB"
                SQL = "CREATE TABLE " & TableViewPrefix & "." & TableName & vbCrLf
                SQL = SQL & "(" & vbCrLf
                SQL = SQL & "TABLE_NAME VARCHAR2(50) NOT NULL," & vbCrLf
                SQL = SQL & "USER_NAME VARCHAR2(20) NOT NULL," & vbCrLf
                SQL = SQL & "ACTION_TYPE VARCHAR2(1) NOT NULL," & vbCrLf
                SQL = SQL & "AUDIT_TIMESTAMP DATE NOT NULL," & vbCrLf
                SQL = SQL & "RECORD_ID NUMBER(6,0) NOT NULL" & vbCrLf
                SQL = SQL & ")" & vbCrLf
                SQL = SQL & "TABLESPACE SYSTEM"
                Call ExecuteDatasourceSQL(DatasourceID, SQL)
            Case "SQLSERVER.OLEDB"
                Set tDef = New ADOX.Table
                tDef.Name = TableName
                tDef.Columns.Append "TABLE_NAME", adVarChar, 50
                tDef.Columns.Append "USER_NAME", adVarChar, 20
                tDef.Columns.Append "ACTION_TYPE", adVarChar, 1
                tDef.Columns.Append "AUDIT_TIMESTAMP", adDBTimeStamp
                tDef.Columns.Append "RECORD_ID", adInteger
                dDef.Tables.Append tDef
        End Select

        Err.Clear
    End If
End Sub
```



**EXHIBIT E**  
**GetPrimaryKey**

```
Private Function GetPrimaryKey(tDef As ADOX.Table) As String
    Dim iDef As ADOX.Index

    GetPrimaryKey = tDef.Columns(0).Name
    For Each iDef In tDef.Indexes
        If iDef.PrimaryKey Then
            GetPrimaryKey = iDef.Columns(0).Name
            Exit For
        End If
    Next iDef
End Function
```

## EXHIBIT E

### GetRecordsource

```
Private Function GetRecordsource(dDef As ADOX.Catalog, TableName As String) As String
    Dim tDef As ADOX.Table
    Dim ViewTagType As String
    Dim ViewTag As String
    Dim ViewName As String

    GetRecordsource = TableName
    ViewTagType = UCase$(GetINISetting(App.Path & "\SystemConsole.ini", "System Settings", "View Tag Type",
"ERROR"))
    ViewTag = GetINISetting(App.Path & "\SystemConsole.ini", "System Settings", "View Tag", "ERROR")

    Select Case ViewTagType
        Case "PREFIX"
            ViewName = ViewTag & TableName
        Case "SUFFIX"
            ViewName = TableName & ViewTag
    End Select
    On Error Resume Next
    Set tDef = dDef.Tables(ViewName)
    If Err.Number = 0 Then
        GetRecordsource = ViewName
    End If
    Err.Clear
End Function
```

**EXHIBIT E**  
**IsReferenceTables**

```
Private Function IsReferenceTables(TableName As String) As Boolean
    Dim ArraySize As Long
    Dim i As Long
    On Error Resume Next
    ArraySize = UBound(gData.ReferenceTables)
    If Err.Number <> 0 Then
        Err.Clear
        IsReferenceTables = False
        Exit Function
    End If
    For i = 0 To ArraySize
        If gData.ReferenceTables(i) = TableName Then
            IsReferenceTables = True
            Exit Function
        End If
    Next i
End Function
```

## EXHIBIT E

### CreatePermissions

```
Public Sub CreatePermissions()  
    Dim rsEdit As ADODB.Recordset  
    Dim rs As ADODB.Recordset  
    Dim GroupID As Variant  
    Dim SQL As String  
  
    On Error Resume Next  
    GroupID = DLookup("GROUP_ID", "FW_GROUP", "GROUP_DESCRIPTION='Administrators'")  
    If IsNull(GroupID) Then Exit Sub  
  
    If PrimaryKeyID = -1 Then  
        Set rs = GetEditRS("Select * From " & m_CurrentTable & " Where " & m_PrimaryKey & "=-1")  
        rs.AddNew  
        rs(m_ForeignKey).Value = CurrentID  
        rs("GROUP_ID").Value = GroupID  
        rs("MENU_CONTEXT_ID").Value = m_MenuContextID  
        rs.Update  
        PrimaryKeyID = rs(m_PrimaryKey).Value  
    End If  
  
    SQL = "Select * From MENU_GROUPING_V Where DEFAULT_PERMISSION=TRUE AND MENU_CONTEXT='" & m_MenuContext & "'"&  
    Set rs = GetRS(SQL)  
    SQL = "Select * From " & m_PermissionTable & " Where " & m_PrimaryKey & "=" & PrimaryKeyID  
    Set rsEdit = GetEditRS(SQL)  
    Do While Not rs.EOF  
        rsEdit.AddNew  
        rsEdit(m_PrimaryKey).Value = PrimaryKeyID  
        rsEdit("MENU_ITEM_GROUPING_ID").Value = rs("MENU_ITEM_GROUPING_ID").Value  
        rsEdit.Update  
        rs.MoveNext  
    Loop  
    rs.Close: Set rs = Nothing  
    rsEdit.Close: Set rsEdit = Nothing  
  
    Err.Clear  
End Sub
```

## EXHIBIT E

### ModifyRelationships

```
Private Sub ModifyRelationships(EntityID As Long, IsReferenceTable As Boolean)
    Dim rs As ADODB.Recordset
    Dim v As Boolean
    Dim cPerm As CPermission
    Dim DatasourceID As Long
    Dim SQL As String

    DatasourceID = DLookup("DATASOURCE_ID", "FW_ENTITY", "ENTITY_ID=" & EntityID)
    Set rs = GetEditRS("Select * From FW_ENTITY_RELATION Where ENTITY_ID=" & EntityID)
    Do While Not rs.EOF
        v = DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & rs("RELATED_ENTITY_ID").Value)
        If IsReferenceTable Or v Then
            rs("RELATIVE_ORDER").Value = Null
            rs("REQUIRED").Value = False
            rs("DISPLAY_ON_TAB").Value = False
            rs("VISIBLE").Value = False
        Else
            rs("VISIBLE").Value = True
        End If
        rs.Update
        SQL = "Delete From FW_RELATION_PERMISSION Where RELATION_GROUP_ID In (Select RELATION_GROUP_ID "
        SQL = SQL & "From FW_RELATION_GROUP Where RELATION_ID=" & rs("RELATION_ID").Value & ")"
        Call ExecutesSQL(SQL)

        SQL = "Delete From FW_RELATION_COMMAND_PERMISSION Where RELATION_GROUP_ID In (Select RELATION_GROUP_ID "
        SQL = SQL & "From FW_RELATION_GROUP Where RELATION_ID=" & rs("RELATION_ID").Value & ")"
        Call ExecutesSQL(SQL)

        SQL = "Delete From FW_RELATION_GROUP Where RELATION_ID=" & rs("RELATION_ID").Value
        Call ExecutesSQL(SQL)

        Set cPerm = New CPermission
        cPerm.PrimaryKeyID = -1
        cPerm.DatasourceID = DatasourceID
        cPerm.CurrentID = rs("RELATION_ID").Value
        cPerm.MenuContext = rs("RELATION_TYPE").Value
        cPerm.CreatePermissions
        Set cPerm = Nothing

        rs.MoveNext
    Loop
    rs.Close: Set rs = Nothing

    Set rs = GetEditRS("Select * From FW_ENTITY_RELATION Where RELATED_ENTITY_ID=" & EntityID)
    Do While Not rs.EOF
        v = DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & rs("ENTITY_ID").Value)
        If IsReferenceTable Or v Then
            rs("RELATIVE_ORDER").Value = Null
            rs("REQUIRED").Value = False
            rs("DISPLAY_ON_TAB").Value = False
            rs("VISIBLE").Value = False
        Else
            rs("VISIBLE").Value = True
        End If
        rs.Update
        SQL = "Delete From FW_RELATION_PERMISSION Where RELATION_GROUP_ID In (Select RELATION_GROUP_ID "
        SQL = SQL & "From FW_RELATION_GROUP Where RELATION_ID=" & rs("RELATION_ID").Value & ")"
        Call ExecutesSQL(SQL)
        SQL = "Delete From FW_RELATION_COMMAND_PERMISSION Where RELATION_GROUP_ID In (Select RELATION_GROUP_ID "
        SQL = SQL & "From FW_RELATION_GROUP Where RELATION_ID=" & rs("RELATION_ID").Value & ")"
        Call ExecutesSQL(SQL)
        SQL = "Delete From FW_RELATION_GROUP Where RELATION_ID=" & rs("RELATION_ID").Value
        Call ExecutesSQL(SQL)
        Set cPerm = New CPermission
        cPerm.PrimaryKeyID = -1
        cPerm.DatasourceID = DatasourceID
        cPerm.CurrentID = rs("RELATION_ID").Value
        cPerm.MenuContext = rs("RELATION_TYPE").Value
        cPerm.CreatePermissions
        Set cPerm = Nothing
        rs.MoveNext
    Loop
    rs.Close: Set rs = Nothing
```

End Sub

## EXHIBIT E

### OrderEntities

```
Private Sub OrderEntities()  
    Dim rs As ADODB.Recordset  
    Dim SQL As String  
    Dim RelativeOrder As Long  
    Dim DefaultIcon As String  
  
    SQL = "Select RELATIVE_ORDER From FW_ENTITY Where Not REFERENCE And Not RELATIVE_ORDER Is Null "  
    SQL = SQL & "Order By RELATIVE_ORDER Desc"  
    RelativeOrder = 1  
    Set rs = GetRS(SQL)  
    If Not rs.EOF And rs.EOF Then  
        RelativeOrder = rs("RELATIVE_ORDER").Value + 1  
    End If  
    rs.Close: Set rs = Nothing  
    DefaultIcon = GetINISetting(App.Path & "\SystemConsole.ini", "System Settings", "Default Entity Icon",  
"ERROR")  
  
    SQL = "Select * From FW_ENTITY Where DISPLAY_ON_MENU And Not REFERENCE And RELATIVE_ORDER Is Null "  
    SQL = SQL & "Order By ENTITY_CAPTION Asc"  
    Set rs = GetEditRS(SQL)  
    Do While Not rs.EOF  
        rs("RELATIVE_ORDER").Value = RelativeOrder  
        If IsNull(rs("ICON").Value) Then  
            rs("ICON").Value = DefaultIcon  
        End If  
        rs.Update  
        RelativeOrder = RelativeOrder + 1  
        rs.MoveNext  
    Loop  
    rs.Close: Set rs = Nothing  
End Sub
```

**EXHIBIT E**  
CreateRelationships

```
Public Sub CreateRelationships()  
    Dim rs As ADODB.Recordset  
  
    Set rs = GetRS("Select ENTITY_ID From FW_ENTITY")  
    Do While Not rs.EOF  
        RefreshRelationships rs("ENTITY_ID").Value  
        rs.MoveNext  
    Loop  
    rs.Close: Set rs = Nothing  
  
End Sub
```

## EXHIBIT E

### RefreshRelationships

```
Public Sub RefreshRelationships(EntityID As Long)
    Dim DatasourceID As Long
    Dim TableName As String
    Dim RelatedEntityID As Long
    Dim SQL As String
    Dim rs As ADODB.Recordset
    Dim con As ADODB.Connection
    Dim TableViewPrefix As String

    DatasourceID = DLookup("DATASOURCE_ID", "FW_ENTITY", "ENTITY_ID=" & EntityID)
    TableName = DLookup("TABLE_NAME", "FW_ENTITY", "ENTITY_ID=" & EntityID)

    Set con = GetDatasourceConnection(DatasourceID)
    If con Is Nothing Then
        Exit Sub
    End If

    TableViewPrefix = DLookup("DATASOURCE_ATTRIBUTE_VALUE", "DATASOURCE_ATTRIBUTE V",
    "DATASOURCE_ATTRIBUTE='INITIAL_CATALOG' AND DATASOURCE_ID=" & DatasourceID) & ""

    Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
        Case "ACCESS.OLEDB"
            Set rs = con.OpenSchema(adSchemaForeignKeys, Array(Empty, Empty, Empty, Empty, Empty, Empty, TableName))
        Case "ORACLE.OLEDB"
            Set rs = con.OpenSchema(adSchemaForeignKeys, Array(Empty, TableViewPrefix, Empty, Empty, Empty, Empty,
            TableName))
        Case "SQLSERVER.OLEDB"
            Set rs = con.OpenSchema(adSchemaForeignKeys, Array(TableViewPrefix, Empty, Empty, Empty, Empty, Empty,
            TableName))
    End Select

    Do While Not rs.EOF
        RelatedEntityID = DLookup("ENTITY_ID", "FW_ENTITY", "TABLE_NAME='" & rs("PK_TABLE_NAME").Value & "'")

        SQL = "Delete From FW_ENTITY_RELATION Where ENTITY_ID=" & EntityID & " And "
        SQL = SQL & "RELATED_ENTITY_ID=" & RelatedEntityID & " And ENTITY_FIELD_NAME="
        SQL = SQL & rs("FK_COLUMN_NAME").Value & "' And RELATED_FIELD_NAME=" & rs("PK_COLUMN_NAME").Value & "'"
        Call ExecutesSQL(SQL)

        SQL = "Delete From FW_ENTITY_RELATION Where ENTITY_ID=" & RelatedEntityID & " And "
        SQL = SQL & "RELATED_ENTITY_ID=" & EntityID & " And ENTITY_FIELD_NAME="
        SQL = SQL & rs("PK_COLUMN_NAME").Value & "' And RELATED_FIELD_NAME=" & rs("FK_COLUMN_NAME").Value & "'"
        Call ExecutesSQL(SQL)

        If rs("FK_TABLE_NAME").Value <> rs("PK_TABLE_NAME").Value Then
            AddRelationship rs("FK_TABLE_NAME").Value, rs("PK_TABLE_NAME").Value, rs("FK_COLUMN_NAME").Value,
            rs("PK_COLUMN_NAME").Value
        End If
        rs.MoveNext
    Loop
    rs.Close: Set rs = Nothing
End Sub
```



## EXHIBIT E

### AddRelationship

```
Private Sub AddRelationship(TableName As String, RelatedTableName As String, ColumnName As String,
RelatedColumnName As String)
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim EntityID As Long
    Dim RelatedEntityID As Long
    Dim ReferenceTable As Boolean
    Dim RelatedReferenceTable As Boolean
    Dim cPerm As CPermission
    Dim RelationID As Long
    Dim DatasourceID As Long

    EntityID = DLookup("ENTITY_ID", "FW_ENTITY", "TABLE_NAME='" & TableName & "'")
    ReferenceTable = DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & EntityID)
    DatasourceID = DLookup("DATASOURCE_ID", "FW_ENTITY", "ENTITY_ID=" & EntityID)
    RelatedEntityID = DLookup("ENTITY_ID", "FW_ENTITY", "TABLE_NAME='" & RelatedTableName & "'")
    RelatedReferenceTable = DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & RelatedEntityID)

    SQL = "Select * From FW_ENTITY_RELATION Where RELATION_TYPE='MANY_TO_ONE' And ENTITY_ID=" & EntityID & " And
RELATED_ENTITY_ID=" & RelatedEntityID
    Set rs = GetEditRS(SQL)
    If rs.BOF And rs.EOF Then
        rs.AddNew
        rs("ENTITY_ID").Value = EntityID
        rs("ENTITY_FIELD_NAME").Value = ColumnName
        rs("RELATED_ENTITY_ID").Value = RelatedEntityID
        rs("RELATED_FIELD_NAME").Value = RelatedColumnName
        rs("RELATION_TYPE").Value = "MANY_TO_ONE"
        rs("RELATIVE_ORDER").Value = Null
        rs("REQUIRED").Value = False
        rs("DISPLAY_ON_TAB").Value = False
        rs("VISIBLE").Value = Not (ReferenceTable Or RelatedReferenceTable)
        rs.Update
        RelationID = rs("RELATION_ID").Value
    End If
    rs.Close: Set rs = Nothing

    Set cPerm = New CPermission
    cPerm.PrimaryKeyID = -1
    cPerm.DatasourceID = DatasourceID
    cPerm.CurrentID = RelationID
    cPerm.MenuContext = "MANY_TO_ONE"
    cPerm.CreatePermissions
    Set cPerm = Nothing

    SQL = "Select * From FW_ENTITY_RELATION Where RELATION_TYPE='ONE_TO_MANY' And ENTITY_ID=" & RelatedEntityID &
" And RELATED_ENTITY_ID=" & EntityID
    Set rs = GetEditRS(SQL)
    If rs.BOF And rs.EOF Then
        rs.AddNew
        rs("ENTITY_ID").Value = RelatedEntityID
        rs("ENTITY_FIELD_NAME").Value = RelatedColumnName
        rs("RELATED_ENTITY_ID").Value = EntityID
        rs("RELATED_FIELD_NAME").Value = ColumnName
        rs("RELATION_TYPE").Value = "ONE_TO_MANY"
        rs("RELATIVE_ORDER").Value = Null
        rs("REQUIRED").Value = False
        rs("DISPLAY_ON_TAB").Value = False
        rs("VISIBLE").Value = Not (ReferenceTable Or RelatedReferenceTable)
        rs.Update
        RelationID = rs("RELATION_ID").Value
    End If
    rs.Close: Set rs = Nothing

    Set cPerm = New CPermission
    cPerm.PrimaryKeyID = -1
    cPerm.DatasourceID = DatasourceID
    cPerm.CurrentID = RelationID
    cPerm.MenuContext = "ONE_TO_MANY"
    cPerm.CreatePermissions
    Set cPerm = Nothing
End Sub
```

**EXHIBIT E**  
**RefreshEntities**

```
Private Sub RefreshEntities()  
    Dim rs As ADODB.Recordset  
  
    Set rs = GetRS("Select ENTITY_ID From FW_ENTITY")  
    Do While Not rs.EOF  
        RefreshEntity rs("ENTITY_ID").Value  
        OrderRelationships rs("ENTITY_ID").Value  
        rs.MoveNext  
    Loop  
    rs.Close: Set rs = Nothing  
End Sub
```

## EXHIBIT E

### RefreshEntity

```
Public Sub RefreshEntity(EntityID As Long)
    Dim i As Long
    Dim fDef As ADOX.Column
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim EntityColumnID As Long
    Dim dDef As ADOX.Catalog
    Dim tDef As ADOX.Table
    Dim PrimaryKey As String
    Dim CallCreateComboBox As Boolean
    Dim Criteria As String
    Dim TableName As String
    Dim ReferenceTable As Boolean

    Set dDef = New ADOX.Catalog
    Set dDef.ActiveConnection = GetDatasourceConnection(DLookup("DATASOURCE_ID", "FW_ENTITY", "ENTITY_ID=" &
EntityID))
    If dDef.ActiveConnection Is Nothing Then
        Set dDef = Nothing
        Exit Sub
    End If
    TableName = DLookup("TABLE_NAME", "FW_ENTITY", "ENTITY_ID=" & EntityID)
    ReferenceTable = DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & EntityID)

    Set tDef = dDef.Tables(TableName)

    PrimaryKey = GetPrimaryKey(tDef)

    For i = 0 To tDef.Columns.Count - 1
        Set fDef = tDef.Columns(i)
        SQL = "Select * From FW_ENTITY_COLUMN Where FIELD_NAME='" & fDef.Name
        SQL = SQL & "' And ENTITY_ID=" & EntityID

        CallCreateComboBox = False
        Set rs = GetEditRS(SQL)
        If rs.BOF And rs.EOF Then
            CallCreateComboBox = True
            rs.AddNew
            rs("ENTITY_ID").Value = EntityID
            rs("FIELD_NAME").Value = fDef.Name
            rs("CONTROL_CAPTION").Value = StrConv(Replace(fDef.Name, "_", " ", 1, -1, vbTextCompare), vbProperCase)
            If Right$(rs("CONTROL_CAPTION").Value, 3) = " Id" Then
                rs("CONTROL_CAPTION").Value = Left$(rs("CONTROL_CAPTION").Value, Len(rs("CONTROL_CAPTION").Value) - 3) &
" ID"
            End If
            Select Case fDef.Type
                Case adBoolean
                    rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE", "CONTROL_TYPE='CheckBox'")
                    rs("DEFAULT_VALUE").Value = Null
                Case adCurrency
                    rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='NumericFormatBox'")
                    rs("COLUMN_FORMAT").Value = "Currency"
                    rs("DEFAULT_VALUE").Value = 0
                Case adBigInt, adTinyInt, adSmallInt, adInteger, adUnsignedTinyInt, adUnsignedSmallInt, adUnsignedInt
                    rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='NumericFormatBox'")
                    rs("COLUMN_FORMAT").Value = "0"
                    rs("DEFAULT_VALUE").Value = Null
                Case Else
                    Select Case DLookup("GENERIC_CONSTANT", "FW_DATATYPE", "ADO_DATA_TYPE=" & fDef.Type)
                        Case "GenericDate"
                            rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='DateTimeControl'")
                        Case "GenericMemo"
                            rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='NotesControl'")
                        Case "GenericNumber"
                            rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='NumericFormatBox'")
                        Case Else
                            rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='TextBoxControl'")
                    End Select
                    rs("DEFAULT_VALUE").Value = Null
                End Select
            End Select
        End If
    Next i
End Sub
```

## EXHIBIT E

```
If PrimaryKey = fDef.Name Then
    rs("TAB_INDEX").Value = 1
    rs("REQUIRED").Value = True
Else
    rs("TAB_INDEX").Value = Null
    Criteria = "RELATED_ENTITY_ID=" & EntityID & " And RELATED_FIELD_NAME='" & fDef.Name & "'"
    If ReferenceTable Then
        rs("REQUIRED").Value = True
    Else
        rs("REQUIRED").Value = Not IsNull(DLookup("RELATION_ID", "FW_ENTITY_RELATION", Criteria))
    End If
End If
rs("VISIBLE").Value = True
rs.Update
EntityColumnID = rs("ENTITY_COLUMN_ID").Value

Select Case fDef.Type
    Case adBoolean
    Case adCurrency
        SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
        SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='DECIMALS'") & "," & '2')'"
        Call ExecutesSQL(SQL)
    Case adBigInt, adTinyInt, adSmallInt, adInteger, adUnsignedTinyInt, adUnsignedSmallInt, adUnsignedInt
        SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
        SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='DECIMALS'") & "," & '0')'"
        Call ExecutesSQL(SQL)
    Case Else
        Select Case DLookup("GENERIC_CONSTANT", "FW_DATATYPE", "ADO_DATA_TYPE=" & fDef.Type)
            Case "GenericDate"
                SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
                SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='DATE_TIME_CONTROL_TYPE'") & "," & 'DATE_CONTROL')'"
                Call ExecutesSQL(SQL)
                SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
                SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='MIN_DATE'") & "," & '01/01/1601')'"
                Call ExecutesSQL(SQL)
                SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
                SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='MAX_DATE'") & "," & '12/31/9999')'"
                Call ExecutesSQL(SQL)
            Case "GenericMemo"
                SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
                SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='LINES'") & "," & '5')'"
                Call ExecutesSQL(SQL)
            Case "GenericNumber"
                SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
                SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='DECIMALS'") & "," & '6')'"
                Call ExecutesSQL(SQL)
            Case Else
                SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE
(ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE) Values("
                SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='LINES'") & "," & '1')'"
                Call ExecutesSQL(SQL)
        End Select
    End Select
End If
rs.Close: Set rs = Nothing
If CallCreateComboBox Then
    CreateComboBoxControls EntityID, fDef.Name
End If
Next i
OrderEntityColumns EntityID
End Sub
```

## EXHIBIT E

### CreateComboBoxControls

```
Private Sub CreateComboBoxControls(EntityID As Long, EntityFieldName As String)
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim ViewColumns As Long
    Dim Recordsource As String

    SQL = "Select * From ENTITY_RELATIONSHIP_V Where RELATION_TYPE='MANY_TO_ONE' "
    SQL = SQL & "And ENTITY_ID=" & EntityID & " And ENTITY_FIELD_NAME='" & EntityFieldName & "'"
    Set rs = GetRS(SQL)
    If Not (rs.BOF And rs.EOF) Then
        If DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & rs("RELATED_ENTITY_ID").Value) Then
            Recordsource = BuildRecordsource(rs("RELATED_ENTITY_ID").Value, ViewColumns)
            CreateComboBoxControl rs("ENTITY_ID").Value, rs("ENTITY_FIELD_NAME").Value, Recordsource, ViewColumns,
rs("RELATED_ENTITY_ID").Value
        Else
            CreateSmartComboControl rs("ENTITY_ID").Value, rs("ENTITY_FIELD_NAME").Value,
rs("RELATED_ENTITY_ID").Value
        End If
    End If
    rs.Close: Set rs = Nothing
End Sub
```

## EXHIBIT E

### BuildRecordsource

```
Private Function BuildRecordsource(EntityID As Long, ViewColumns As Long) As String
    Dim SQL As String
    Dim ViewName As String
    Dim rs As ADODB.Recordset
    Dim DatasourceID As Long
    Dim TableViewPrefix As String

    ViewName = DLookup("RECORD_SOURCE", "FW_ENTITY", "ENTITY_ID=" & EntityID)
    DatasourceID = DLookup("DATASOURCE_ID", "FW_ENTITY", "ENTITY_ID=" & EntityID)

    TableViewPrefix = DLookup("DATASOURCE_ATTRIBUTE_VALUE", "DATASOURCE_ATTRIBUTE_V",
"DATASOURCE_ATTRIBUTE='INITIAL_CATALOG' AND DATASOURCE_ID=" & DatasourceID) & ""

    Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
        Case "ORACLE.OLEDB"
            ViewName = TableViewPrefix & "." & ViewName
        Case Else
    End Select

    Set rs = GetDatasourceRS(DatasourceID, "Select * From " & ViewName & " Where 1=2")

    If rs.Fields.Count > 1 Then
        SQL = "SELECT " & rs(0).Name & ", " & rs(1).Name & " FROM " & ViewName & " ORDER BY " & rs(1).Name
        ViewColumns = 2
    Else
        SQL = "SELECT " & rs(0).Name & " FROM " & ViewName & " ORDER BY " & rs(0).Name
        ViewColumns = 1
    End If
    rs.Close: Set rs = Nothing
    BuildRecordsource = SQL
End Function
```

## EXHIBIT E

### CreateComboBoxControl

```
Private Sub CreateComboBoxControl(EntityID As Long, FieldName As String, Recordsource As String, ViewColumns As Long, RelatedEntityID As Long)
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim EntityColumnID As Long

    SQL = "Select * From FW_ENTITY_COLUMN Where ENTITY_ID=" & EntityID & " And FIELD_NAME='"
    SQL = SQL & FieldName & "'"
    Set rs = GetEditRS(SQL)
    EntityColumnID = -1
    If Not (rs.EOF And rs.BOF) Then
        rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='ComboBoxControl'")
        If UCase$(Right$(rs("CONTROL_CAPTION").Value, 3)) = " ID" Then
            rs("CONTROL_CAPTION").Value = Left$(rs("CONTROL_CAPTION").Value, Len(rs("CONTROL_CAPTION").Value) - 3)
        End If
        EntityColumnID = rs("ENTITY_COLUMN_ID").Value
        rs.Update
    End If
    If EntityColumnID = -1 Then Exit Sub

    SQL = "Delete From FW_COLUMN_ATTRIBUTE_VALUE Where ENTITY_COLUMN_ID=" & EntityColumnID

    SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE (ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE)
Values("
    SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='RECORD_SOURCE'") & "," & Recordsource & "'"
    Call ExecuteSQL(SQL)
    SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE (ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE)
Values("
    SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='ALLOW_EDIT'") & "," & "0'"
    Call ExecuteSQL(SQL)
    SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE (ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE)
Values("
    SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='DISPLAY_COLUMN'") & "," & ViewColumns - 1 & "'"
    Call ExecuteSQL(SQL)
    SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE (ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE)
Values("
    SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='REFERENCE_ENTITY_ID'") & "," & RelatedEntityID & "'"
    Call ExecuteSQL(SQL)

End Sub
```

**EXHIBIT E**  
CreateSmartComboControl

```
Private Sub CreateSmartComboControl(EntityID As Long, FieldName As String, RelatedEntityID As Long)
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim EntityColumnID As Long

    SQL = "Select * From FW_ENTITY_COLUMN Where ENTITY_ID=" & EntityID & " And FIELD_NAME='"
    SQL = SQL & FieldName & "'"
    Set rs = GetEditRS(SQL)
    EntityColumnID = -1
    If Not (rs.BOF And rs.EOF) Then
        rs("CONTROL_TYPE_ID").Value = DLookup("CONTROL_TYPE_ID", "FW_CONTROL_TYPE",
"CONTROL_TYPE='SmartComboControl'")
        EntityColumnID = rs("ENTITY_COLUMN_ID").Value
        rs.Update
    End If
    If EntityColumnID = -1 Then Exit Sub

    SQL = "Delete From FW_COLUMN_ATTRIBUTE_VALUE Where ENTITY_COLUMN_ID=" & EntityColumnID

    SQL = "Insert Into FW_COLUMN_ATTRIBUTE_VALUE (ENTITY_COLUMN_ID,COLUMN_ATTRIBUTE_ID,COLUMN_ATTRIBUTE_VALUE)
Values("
    SQL = SQL & EntityColumnID & "," & DLookup("COLUMN_ATTRIBUTE_ID", "FW_COLUMN_ATTRIBUTE",
"COLUMN_ATTRIBUTE='REFERENCE_ENTITY_ID'") & "," & RelatedEntityID & "'"
    Call ExecuteSQL(SQL)
End Sub
```



**EXHIBIT E**  
OrderEntityColumns

```
Private Sub OrderEntityColumns(EntityID As Long)
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim TabIndex As Long
    Dim DefaultIcon As String

    SQL = "Select TAB_INDEX From FW_ENTITY_COLUMN Where Not TAB_INDEX Is Null And ENTITY_ID=" & EntityID
    SQL = SQL & " Order By TAB_INDEX Desc"
    TabIndex = 1
    Set rs = GetRS(SQL)
    If Not (rs.BOF And rs.EOF) Then
        TabIndex = rs("TAB_INDEX").Value + 1
    End If
    rs.Close: Set rs = Nothing

    SQL = "Select * From FW_ENTITY_COLUMN Where TAB_INDEX Is Null And ENTITY_ID=" & EntityID
    SQL = SQL & " Order By FIELD_NAME Asc"
    Set rs = GetEditRS(SQL)
    Do While Not rs.EOF
        rs("TAB_INDEX").Value = TabIndex
        rs.Update
        TabIndex = TabIndex + 1
        rs.MoveNext
    Loop
    rs.Close: Set rs = Nothing
End Sub
```

## EXHIBIT E

### OrderRelationships

```
Public Sub OrderRelationships(EntityID As Long)
    Dim rs As ADODB.Recordset
    Dim SQL As String
    Dim RelativeOrder As Long

    If DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & EntityID) Then Exit Sub

    SQL = "Select RELATIVE_ORDER From ENTITY_RELATIONSHIP_V Where Not REFERENCE And Not RELATIVE_ORDER Is Null "
    SQL = SQL & "And ENTITY_ID=" & EntityID & " Order By RELATIVE_ORDER Desc"
    RelativeOrder = 1
    Set rs = GetRS(SQL)
    If Not rs.BOF And rs.EOF Then
        RelativeOrder = rs("RELATIVE_ORDER").Value + 1
    End If
    rs.Close: Set rs = Nothing

    SQL = "Select * From FW_ENTITY_RELATION Where RELATIVE_ORDER Is Null And ENTITY_ID=" & EntityID
    Set rs = GetEditRS(SQL)
    Do While Not rs.EOF
        If Not DLookup("REFERENCE", "FW_ENTITY", "ENTITY_ID=" & rs("RELATED_ENTITY_ID").Value) Then
            rs("RELATIVE_ORDER").Value = RelativeOrder
            rs.Update
            RelativeOrder = RelativeOrder + 1
        End If
        rs.MoveNext
    Loop
    rs.Close: Set rs = Nothing
End Sub
```

## **EXHIBIT E**

### LoadSequences

```
Private Sub LoadSequences()  
    Dim rs As ADODB.Recordset  
    Dim NextVal As Long  
  
    Set rs = GetRS("Select * From FW_ENTITY")  
    Do While Not rs.EOF  
        NextVal = GetNextSequenceVal(rs("DATASOURCE_ID").Value, rs("TABLE_NAME").Value,  
rs("UNIQUE_FIELD_NAME").Value)  
        InsertSequence rs("DATASOURCE_ID").Value, rs("TABLE_NAME").Value, NextVal  
        rs.MoveNext  
    Loop  
    rs.Close: Set rs = Nothing  
End Sub
```

## EXHIBIT E

### InsertSequence

```
Private Sub InsertSequence(DatasourceID As Long, TableName As String, NextVal As Long)
    Dim rs As ADODB.Recordset
    Dim com As ADODB.Command
    Dim TableViewPrefix As String

    Set com = New ADODB.Command

    Set com.ActiveConnection = GetDatasourceConnection(DatasourceID)
    If com.ActiveConnection Is Nothing Then
        Set com = Nothing
        Exit Sub
    End If
    TableViewPrefix = DLookup("DATASOURCE_ATTRIBUTE_VALUE", "DATASOURCE_ATTRIBUTE_V",
"DATASOURCE_ATTRIBUTE='INITIAL_CATALOG' AND DATASOURCE_ID=" & DatasourceID) & ""

    On Error Resume Next
    com.CommandType = adCmdText

    Select Case DLookup("Platform", "qryDatasources", "DATASOURCE_ID=" & DatasourceID)
        Case "ACCESS.OLEDB", "SQLSERVER.OLEDB"
            com.CommandText = "Select * From SEQUENCE Where TABLE_NAME='" & TableName & "'"
        Case "ORACLE.OLEDB"
            com.CommandText = "Select * From " & TableViewPrefix & ".SEQUENCE Where TABLE_NAME='" & TableName & "'"
    End Select

    com.CommandTimeout = 300

    Set rs = New ADODB.Recordset
    rs.CursorType = adOpenStatic
    rs.CursorLocation = adUseClient
    rs.LockType = adLockOptimistic
    rs.Open com
    If Err.Number = 0 Then
        If rs.BOF And rs.EOF Then
            rs.AddNew
        End If
        rs("TABLE_NAME").Value = TableName
        rs("NEXT_VAL").Value = NextVal
        rs.Update
    End If
    Err.Clear
    Set com = Nothing
    rs.Close: Set rs = Nothing
End Sub
```

**EXHIBIT E**  
**RefreshContexts**

```
Private Sub RefreshContexts()  
    Dim rs As ADODB.Recordset  
  
    Set rs = GetRS("Select ENTITY_ID From FW_ENTITY")  
    Do While Not rs.EOF  
        DefineContexts rs("ENTITY_ID").Value  
        rs.MoveNext  
    Loop  
    rs.Close: Set rs = Nothing  
End Sub
```

## EXHIBIT E

### DefineContexts

```
Private Sub DefineContexts(EntityID As Long)
    Dim SQL As String
    Dim rs As ADODB.Recordset

    If IsNull(DLookup("ENTITY_CONTEXT_ID", "FW_ENTITY_CONTEXT", "ENTITY_ID=" & EntityID)) Then
        SQL = "Select * From FW_ENTITY_RELATION Where ENTITY_ID=" & EntityID
        Set rs = GetRS(SQL)
        Do While Not rs.EOF
            Select Case rs("RELATION_TYPE").Value
                Case "ONE_TO_MANY", "MANY_TO_MANY"
                    DefineContext rs("ENTITY_ID").Value, rs("RELATED_ENTITY_ID").Value, False
                Case "MANY_TO_ONE"
                    DefineContext rs("ENTITY_ID").Value, rs("RELATED_ENTITY_ID").Value, True
            End Select
            rs.MoveNext
        Loop
        rs.Close: Set rs = Nothing
    End If
End Sub
```

## EXHIBIT E

### DefineContext

```
Private Sub DefineContext(EntityID As Long, RelatedEntityID As Long, ReverseJoin As Boolean)
    Dim SQL As String
    Dim rs As ADODB.Recordset
    Dim rsEdit As ADODB.Recordset

    If ReverseJoin Then
        SQL = "Select TableJoin From qryTableJoins Where ENTITY_ID=" & RelatedEntityID
        SQL = SQL & " And RELATED_ENTITY_ID=" & EntityID
    Else
        SQL = "Select TableJoin From qryTableJoins Where ENTITY_ID=" & EntityID
        SQL = SQL & " And RELATED_ENTITY_ID=" & RelatedEntityID
    End If
    Set rs = GetRS(SQL)
    SQL = "Select * From FW_ENTITY_CONTEXT Where ENTITY_ID=" & EntityID
    Set rsEdit = GetEditRS(SQL)
    Do While Not rs.EOF
        rsEdit.AddNew
        rsEdit("ENTITY_ID").Value = EntityID
        rsEdit("TABLE_JOIN").Value = rs("TableJoin").Value
        rsEdit.Update
        rs.MoveNext
    Loop
    rs.Close: Set rs = Nothing
    rsEdit.Close: Set rsEdit = Nothing
End Sub
```